Better exploiting motion for better action recognition

Mihir Jain, Hervé Jégou, Patrick Bouthemy

Better exploiting motion for better action recognition

Mihir Jain Hervé Jégou Patrick Bouthemy
INRIA, Rennes, France

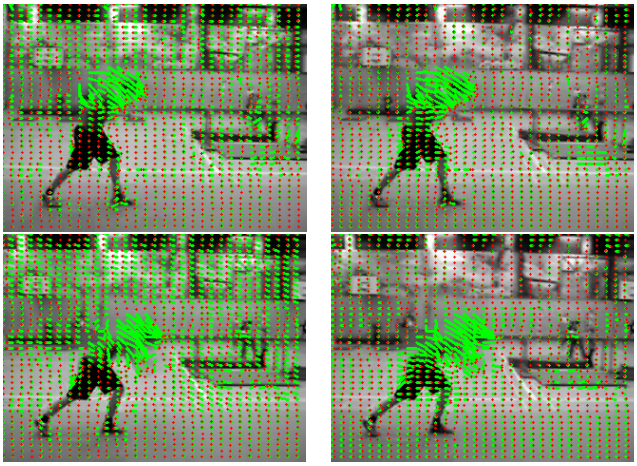
Abstract

Several recent works on action recognition have attested the importance of explicitly integrating motion characteristics in the video description. This paper establishes that adequately decomposing visual motion into dominant and residual motions, both in the extraction of the space-time trajectories and for the computation of descriptors, significantly improves action recognition algorithms. Then, we design a new motion descriptor, the DCS descriptor, based on differential motion scalar quantities, divergence, curl and shear features. It captures additional information on the local motion patterns enhancing results. Finally, applying the recent VLAD coding technique proposed in image retrieval provides a substantial improvement for action recognition. Our three contributions are complementary and lead to outperform all reported results by a significant margin on three challenging datasets, namely Hollywood 2, HMDB51 and Olympic Sports.

1. Introduction and related work

Human actions often convey the essential meaningful content in videos. Yet, recognizing human actions in unconstrained videos is a challenging problem in Computer Vision which receives a sustained attention due to the potential applications. In particular, there is a large interest in designing video-surveillance systems, providing some automatic annotation of video archives as well as improving human-computer interaction. The solutions proposed to address this problem inherit, to a large extent, from the techniques first designed for the goal of image search and classification. The successful local features developed to describe image patches [14, 22] have been translated in the 2D+t domain as spatio-temporal local descriptors [13, 29] and now include motion clues [28]. These descriptors are often extracted from spatial-temporal interest points [12, 30]. More recent techniques assume some underlying temporal motion model involving trajectories [3, 7, 16, 17, 24, 28, 31].

Most of these approaches produce large set of local descriptors which are in turn aggregated to produce a single vector representing the video, in order to enable the use of powerful discriminative classifiers such as support vector machines (SVMs). This is usually done with the bag-of-words technique [23], which quantizes the local features using a k-means codebook. Thanks to the successful combination of this encoding technique with the aforementioned local descriptors, the state of the art in action recognition is able to go beyond the toy problems of classifying simple human actions in controlled environment and considers the detection of actions in real movies or video clips [11, 15]. Despite these progresses, the existing descriptors suffer from an uncompleted handling of motion in the video sequence.



Optical flow vectors Compensated flow vectors

Figure 1. Optical flow field vectors (green vectors with red end points) before and after dominant motion compensation. Most of the flow vectors due to camera motion are suppressed after compensation. One of the contributions of this paper is to show that compensating for the dominant motion is beneficial for most of the existing descriptors used for action recognition.

Motion is arguably the most reliable source of information for action recognition, as often related to the actions of interest. However, it inevitably involves the background or camera motion when dealing with uncontrolled and realistic situations. Although some attempts have been made to compensate camera motion in several ways [10, 20, 25, 28, 31], how to separate action motion from that caused by the camera, and how to reflect it in the video description remains an open issue. The motion compensation mechanism employed in [10] is tailor-made to the Motion Interchange Pattern encoding technique. The Motion Boundary Histogram (MBH) [28] is a recent appealing approach to

suppress the constant motion by considering the flow gradient. It is robust to some extent to the presence of camera motion, yet it does not explicitly handle the camera motion. Another approach [25] uses a sophisticated and robust (RANSAC) estimation of camera motion. It first segments the color image into regions corresponding to planar parts in the scene and estimates the (three) dominant homographies to update the motion associated with local features. A rather different view is adopted in [31] where the motion decomposition is performed at the trajectory level. All these works support the potential of motion compensation.

As the first contribution of this paper, we address the problem in a way that departs from these works by considering the compensation of the dominant motion in *both* the tracking stages and encoding stages involved in the computation of action recognition descriptors. We rely on the pioneering works on motion compensation such as the technique proposed in [19], that considers 2D polynomial affine motion models for estimating the dominant image motion. We consider this particular model for its robustness and its low computational cost. It was already used in [20] to separate the dominant motion (assumed to be due to the camera motion) and the residual motion (corresponding to the independent scene motions) for dynamic event recognition in videos. However, the statistical modeling of both motion components was global (over the entire image) and only the normal flow was computed for the latter.

Figure 1 shows the vectors of optical flow before and after applying the proposed motion compensation. Our method successfully suppresses most of the background motion and reinforces the focus towards the action of interest. We exploit this compensated motion *both* for descriptor computation and for extracting trajectories. However, we also show that the camera motion should not be thrown as it contains complementary information that is worth using to recognize certain action categories.

Then, we introduce the Divergence-Curl-Shear (DCS) descriptor, which encodes scalar first-order motion features, namely the motion divergence, curl and shear. It captures physical properties of the flow pattern that are not involved in the best existing descriptors for action recognition, except in the work of [1] which exploits divergence and vorticity among a set of eleven kinematic features computed from the optical flow. Our DCS descriptor provides a good performance recognition performance on its own. Most importantly, it conveys some information which is not captured by existing descriptors and further improves the recognition performance when combined with the other descriptors.

As a last contribution, we bring an encoding technique known as VLAD (vector of local aggregated descriptors) [8] to the field of action recognition. This technique is shown to be better than the bag-of-words representation for combining all the local video descriptors we have considered.

The organization of the paper is as follows. Section 2 introduces the motion properties that we will consider through this paper. Section 3 presents the datasets and classification scheme used in our different evaluations. Section 4 details how we revisit several popular descriptors of the literature by the means of dominant motion compensation. Our DCS descriptor based on kinematic properties is introduced in Section 5 and improved by the VLAD encoding technique, which is introduced and bench-marked in Section 6 for several video descriptors. Section 7 provides a comparison showing the large improvement achieved over the state of the art. Finally, Section 8 concludes the paper.

2. Motion Separation and Kinematic Features

In this section, we describe the motion clues we incorporate in our action recognition framework. We separate the dominant motion and the residual motion. In most cases, this will account to distinguishing the impact of camera movement and independent actions. Note that we do not aim at recovering the 3D camera motion: The 2D parametric motion model describes the global (or dominant) motion between successive frames. We first explain how we estimate the dominant motion and employ it to separate the dominant flow from the optical flow. Then, we will introduce kinematic features, namely divergence, curl and shear for a more comprehensive description of the visual motion.

2.1. Affine motion for compensating camera motion

Among polynomial motion models, we consider the 2D affine motion model. Simplest motion models such as the 4-parameter model formed by the combination of 2D translation, 2D rotation and scaling, or more complex ones such as the 8-parameter quadratic model (equivalent to a homography), could be selected as well. The affine model is a good trade-off between accuracy and efficiency which is of primary importance when processing a huge video database. It does have limitations since strictly speaking it implies a single plane assumption for the static background. However, this is not that penalizing (especially for outdoor scenes) if differences in depth remain moderated with respect to the distance to the camera. The affine flow vector at point $p = (x, y)$ and at time t, is defined as

$$w_{\text{aff}}(p_t) = \begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} + \begin{bmatrix} a_1(t) & a_2(t) \\ a_3(t) & a_4(t) \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix}. \quad (1)$$

$u_{\text{aff}}(p_t) = c_1(t) + a_1(t)x_t + a_2(t)y_t$ and $v_{\text{aff}}(p_t) = c_2(t) + a_3(t)x_t + a_4(t)y_t$ are horizontal and vertical components of $w_{\text{aff}}(p_t)$ respectively. Let us denote the optical flow vector at point p at time t as $w(p_t) = (u(p_t), v(p_t))$. We introduce the flow vector $\omega(p_t)$ obtained by removing the affine flow vector from the optical flow vector

$$\omega(p_t) = w(p_t) - w_{\text{aff}}(p_t). \quad (2)$$

The dominant motion (estimated as $w_{\text{aff}}(p_t)$) is usually due to the camera motion. In this case, Equation 2 amounts to canceling (or compensating) the camera motion. Note that this is not always true. For example in case of close-up on a moving actor, the dominant motion will be the affine estimation of the apparent actor motion. The interpretation of the motion compensation output will not be that straightforward in this case, however the resulting ω-field will still exhibit different patterns for the foreground action part and the background part. In the remainder, we will refer to the "compensated" flow as ω-*flow*.

Figure 1 displays the computed optical flow and the ω-flow. We compute the affine flow with the publicly available Motion2D software[1] [19] which implements a real-time robust multiresolution incremental estimation framework. The affine motion model has correctly accounted for the motion induced by the camera movement which corresponds to the dominant motion in the image pair. Indeed, we observe that the compensated flow vectors in the background are close to null and the compensated flow in the foreground, i.e., corresponding to the actors, is conversely inflated. The experiments presented along this paper will show that effective separation of dominant motion from the residual motions is beneficial for action recognition. As explained in Section 4, we will compute local motion descriptors, such as HOF, on both the optical flow and the compensated flow (ω-flow), which allows us to explicitly and directly characterize the scene motion.

2.2. Local kinematic features

By kinematic features, we mean local first-order differential scalar quantities computed on the flow field. We consider the divergence, the curl (or vorticity) and the hyperbolic terms. They inform on the physical pattern of the flow so that they convey useful information on actions in videos. They can be computed from the first-order derivatives of the flow at every point p at every frame t as

$$\begin{cases} \text{div}(p_t) & = \frac{\partial u(p_t)}{\partial x} + \frac{\partial v(p_t)}{\partial y} \\ \text{curl}(p_t) & = \frac{-\partial u(p_t)}{\partial y} + \frac{\partial v(p_t)}{\partial x} \\ \text{hyp}_1(p_t) & = \frac{\partial u(p_t)}{\partial x} - \frac{\partial v(p_t)}{\partial y} \\ \text{hyp}_2(p_t) & = \frac{\partial u(p_t)}{\partial y} + \frac{\partial v(p_t)}{\partial x} \end{cases} \tag{3}$$

The divergence is related to axial motion, expansion and scaling effects, the curl to rotation in the image plane. The hyperbolic terms express the shear of the visual flow corresponding to more complex configuration. We take into account the shear quantity only:

$$\text{shear}(p_t) = \sqrt{\text{hyp}_1^2(p_t) + \text{hyp}_2^2(p_t)}. \tag{4}$$

In Section 5, we propose the DCS descriptor that is based on the kinematic features (divergence, curl and shear) of the visual motion discussed in this subsection. It is computed on either the optical or the compensated flow, ω-flow.

3. Datasets and evaluation

This section first introduces the datasets used for the evaluation. Then, we briefly present the bag-of-feature model and the classification scheme used to encode the descriptors which will be introduced in Section 4.

Hollywood2. The Hollywood2 dataset [15] contains 1,707 video clips from 69 movies representing 12 action classes. It is divided into train set and test set of 823 and 884 samples respectively. Following the standard evaluation protocol of this benchmark, we use average precision (AP) for each class and the mean of APs (mAP) for evaluation.

HMDB51. The HMDB51 dataset [11] is a large dataset containing 6,766 video clips extracted from various sources, ranging from movies to YouTube. It consists of 51 action classes, each having at least 101 samples. We follow the evaluation protocol of [11] and use three train/test splits, each with 70 training and 30 testing samples per class. The average classification accuracy is computed over all classes. Out of the two released sets, we use the original set as it is more challenging and used by most of the works reporting results in action recognition.

Olympic Sports. The third dataset we use is Olympic Sports [18], which again is obtained from YouTube. This dataset contains 783 samples with 16 sports action classes. We use the provided[2] train/test split, there are 17 to 56 training samples and 4 to 11 test samples per class. Mean AP is used for the evaluation, which is the standard choice.

Bag of features and classification setup. We first adopt the standard BOF [23] approach to encode all kinds of descriptors. It produces a vector that serves as the video representation. The codebook is constructed for each type of descriptor separately by the k-means algorithm. Following a common practice in the literature [26, 28, 29], the codebook size is set to k=4,000 elements. Note that Section 6 will consider encoding technique for descriptors.

For the classification, we use a non-linear SVM with χ^2-kernel. When combining different descriptors, we simply add the kernel matrices, as done in [26]:

$$K(x_i, x_j) = \exp\left(-\sum_c \frac{1}{\gamma^c} D(x_i^c, x_j^c)\right), \tag{5}$$

[1] http://www.irisa.fr/vista/Motion2D/

[2] http://vision.stanford.edu/Datasets/OlympicSports/

where $D(x_i^c, x_j^c)$ is χ^2 distance between video x_i^c and x_j^c with respect to *c-th* channel, corresponding to *c-th* descriptor. The quantity γ^c is the mean value of χ^2 distances between the training samples for the *c-th* channel. The multi-class classification problem that we consider is addressed by applying a *one-against-rest* approach.

4. Compensated descriptors

This section describes how the compensation of the dominant motion is exploited to improve the quality of descriptors encoding the motion and the appearance around spatio-temporal positions, hence the term "compensated descriptors". First, we briefly review the local descriptors [6, 13, 15, 28, 29] used here along with dense trajectories [28]. Second, we analyze the impact of motion flow compensation when used in two different stages of the descriptor computation, namely in the tracking and the description part.

4.1. Dense trajectories and local descriptors

Employing dense trajectories to compute local descriptors is one of the state-of-the-art approaches for action recognition. It has been shown [28] that when local descriptors are computed over dense trajectories the performance improves considerably compared to when computed over spatio temporal features [29].

Dense Trajectories [28]: The trajectories are obtained by densely tracking sampled points using optical flow fields. First, feature points are sampled from a dense grid, with step size of 5 pixels and over 8 scales. Each feature point $p_t = (x_t, y_t)$ at frame t is then tracked to the next frame by median filtering in a dense optical flow field $F = (u_t, v_t)$ as follows:

$$p_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * F)|_{(\bar{x_t}, \bar{y_t})}, \quad (6)$$

where M is the kernel of median filtering and $(\bar{x_t}, \bar{y_t})$ is the rounded position of (x_t, y_t). The tracking is limited to L (=15) frames to avoid any drifting effect. Excessively short trajectories and trajectories exhibiting sudden large displacements are removed as they induce some artifacts. Trajectories must be understood here as tracks in the space-time volume of the video.

Local descriptors: The descriptors are computed within a space-time volume centered around each trajectory. Four types of descriptors are computed to encode the shape of the trajectory, local motion pattern and appearance, namely Trajectory [28], HOF (histograms of optical flow) [13], MBH [5] and HOG (histograms of oriented gradients) [4]. All these descriptors depend on the flow field used for the tracking and as input of the descriptor computation:

1. The **Trajectory** descriptor encodes the shape of the trajectory represented by the normalized relative coordinates of the successive points forming the trajectory. It directly depends on the dense flow used for tracking points.
2. **HOF** is computed using the orientations and magnitudes of the flow field.
3. **MBH** is designed to capture the gradient of horizontal and vertical components of the flow. The motion boundaries encode the relative pixel motion and therefore suppress camera motion, but only to some extent.
4. **HOG** encodes the appearance by using the intensity gradient orientations and magnitudes. It is formally not a motion descriptor. Yet the position where the descriptor is computed depends on the trajectory shape.

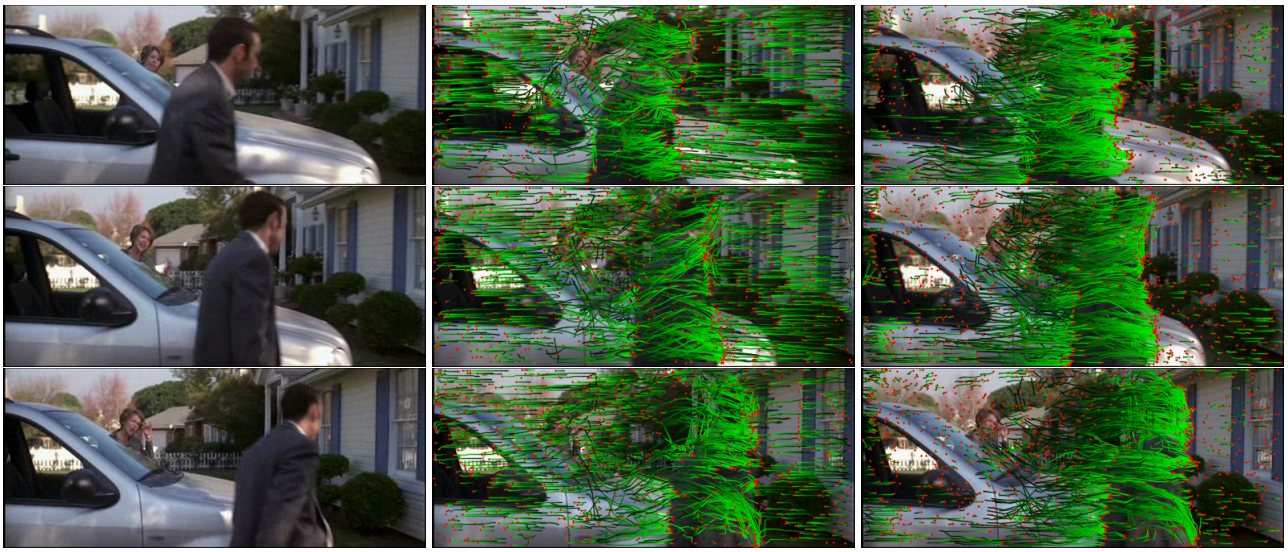
As in [28], volume around a feature point is divided into a $2 \times 2 \times 3$ space-time grid. The orientations are quantized into 8 bins for HOG and 9 bins for HOF (with one additional zero bin). The horizontal and vertical components of MBH are separately quantized into 8 bins each.

4.2. Impact of motion compensation

The optical flow is simply referred to as *flow* in the following, while the compensated flow (see subsection 2.1) is denoted by *ω-flow*. Both of them are considered in the tracking and descriptor computation stages. The trajectories obtained by tracking with the ω-flow are called *ω-trajectories*. Figure 2 comparatively illustrates the ω-trajectories and the trajectories obtained using the flow. The input video shows a man moving away from the car. In this video excerpt, the camera is following the man walking to the right, thus inducing a global motion to the left in the video. When using the flow, the computed trajectories reflect the combination of these two motion components (camera *and* scene motion) as depicted by Subfigure 2(b), which hampers the characterization of the current action. In contrast, the ω-trajectories plotted in Subfigure 2(c) are more active on the actor moving on the foreground, while those localized in the background are now parallel to the time axis enhancing static parts of the scene. The ω-trajectories are therefore more relevant for action recognition, since they are more regularly and more exclusively following the actor's motion.

Impact on Trajectory and HOG descriptors. Table 1 reports the impact of ω-trajectories on Trajectory and HOG descriptors, which are both significantly improved by 3%-4% of mAP on the two datasets. When improved by ω-flow, these descriptors will be respectively referred to as *ω-Trajdesc* and *ω-HOG* in the rest of the paper.

Although the better performance of ω-Trajdesc versus the original Trajectory descriptor was expected, the one



(a) Consecutive frames (b) Trajectories from Optical Flow (c) ω-trajectories

Figure 2. Trajectories obtained from optical and compensated flows. The green tail is the trajectory over 15 frames with red dot indicating the current frame. The trajectories are sub-sampled for the sake of clarity. The frames are extracted every 5 frames in this example.

Descriptor	Hollywood2	HMDB51
Trajectory [28]	47.7%	–
Baseline (reproduced)	47.7%	28.8%
ω-Trajdesc	**51.4%**	**32.9%**
HOG [28]	41.5%	–
Baseline (reproduced)	41.8%	26.3%
ω-HOG	**45.6%**	**29.1%**

Table 1. ω-Trajdesc and ω-HOG: Impact of compensating flow on Trajectory descriptor and HOG descriptors.

Method		Hollywood2	HMDB51
HOF [28]		50.8%	–
HOF (Tracking flow)	flow	50.8%	30.8%
	ω-flow	52.4%	**36.8%**
	both	**54.1%**	**37.7%**
HOF (Tracking ω-flow)	flow	50.2%	33.0%
	ω-flow	52.5%	**37.1%**
	both: ω-HOF	**53.9%**	**38.6%**

Table 2. Impact of using ω-flow on HOF descriptors: mAP for Hollywood2 and average accuracy for HMDB51. The ω-HOF is used in subsequent evaluations.

achieved by ω-HOG might be surprising. Our interpretation is that HOG captures more context with the modified trajectories. More precisely, the original HOG descriptor is computed from a 2D+t sub-volume aligned with the corresponding trajectory and hence represents the appearance along the trajectory shape. When using ω-flow, we do not align the video sequence. As a result, the ω-HOG descriptor is no more computed around the very same tracked physical point in the space-time volume but around points lying in a patch of the initial feature point, whose size depends on the affine flow magnitude. ω-HOG can be viewed as a "patch-based" computation capturing more information about the appearance of the background or of the moving foreground. As for ω-trajectories, they are closer to the real trajectories of the moving actors as they usually cancel the camera movement, and so, more easier to train and recognize.

Impact on HOF. The ω-flow impacts both the trajectory computation used as an input to HOF and the descriptor computation itself. Therefore, HOF can be computed along both types of trajectories (ω-trajectories or those extracted from flow) and can encode both kinds of flows (ω-flow or flow). For the sake of completeness, we evaluate all the variants as well as the combination of both flows in the descriptor computation stage.

The results are presented in Table 2 and demonstrate the significant improvement obtained by computing the HOF descriptor with the ω-flow instead of the optical flow. Note that the type of trajectories which is used, either "Tracking flow" or "Tracking ω-flow", has a limited impact in this case. From now on, we only consider the "Tracking ω-flow" case where HOF is computed along ω-trajectories.

Interestingly, combining the HOF computed from the flow and the ω-flow further improves the results. This suggests that the two flow fields are complementary and the affine flow that was subtracted from ω-flow brings in additional information. For the sake of brevity, the combination of the two kinds of HOF, *i.e.*, computed from the flow and the ω-flow using ω-trajectories, is referred to as the ω-*HOF*

Method		Hollywood2	HMDB51
MBH [28]		54.2%	–
MBH (Tracking *flow*)	flow	**54.2%**	39.7%
	ω-flow	54.0%	39.3%
MBH (Tracking ω-flow)	flow	52.7%	**40.9%**
	ω-flow	52.5%	40.6%

Table 3. Impact of using ω-flow MBH descriptors: mAP for Hollywood2 and average accuracy for HMDB51.

Descriptor	Tracking with	Computing descriptor with	ω-flow descriptor
Trajectory	ω-flow	N/A	ω-Trajdesc
HOG	ω-flow	N/A	ω-HOG
HOF	ω-flow	ω-flow + flow	ω-HOF
MBH	ω-flow	ω-flow	ω-MBH

Table 4. Summary of the updated ω-*flow* descriptors

descriptor in the rest of this paper. Compared to the HOF baseline, the ω-HOF descriptor achieves a gain of $+3.1\%$ of mAP on Hollywood 2 and of $+7.8\%$ on HMDB51.

Impact on MBH. Since MBH is computed from gradient of flow and cancel the constant motion, there is practically no benefit in using the ω-flow to compute the MBH descriptors, as shown in Table 3. However, by tracking ω-flow, the performance improves by around 1.3% for HMDB51 dataset and drops by around 1.5% for Hollywood2. This relative performance depends on the encoding technique. We will come back on this descriptor when considering another encoding scheme for local descriptors in Section 6.

4.3. Summary of compensated descriptors

Table 4 summarizes the refined versions of the descriptors obtained by exploiting the ω-flow, and both ω-flow and the optical flow in the case of HOF. The revisited descriptors considerably improve the results compared to the original ones, with the noticeable exception of ω-MBH which gives mixed performance with a bag-of-features encoding scheme. But we already mention as this point that this incongruous behavior of ω-MBH is stabilized with the VLAD encoding scheme considered in Section 6.

Another advantage of tracking the compensated flow is that fewer trajectories are produced. For instance, the total number of trajectories decreases by about 9.16% and 22.81% on the Hollywood2 and HMDB51 datasets, respectively. Note that exploiting both the flow and the ω-flow do not induce much computational overhead, as the latter is obtained from the flow and the affine flow which is computed in real-time and already used to get the ω-trajectories. The only additional computational cost that we introduce by using the descriptors summarized in Table 4 is the computation of a second HOF descriptor, but this stage is relatively efficient and not the bottleneck of the extraction procedure.

5. Divergence-Curl-Shear descriptor

This section introduces a new descriptor encoding the kinematic properties of motion discussed in Section 2.2. It is denoted by DCS in the rest of this paper.

Combining kinematic features. The spatial derivatives are computed for the horizontal and vertical components of the flow field, which are used in turn to compute the divergence, curl and shear scalar values, see Equation 3.

We consider all possible pairs of kinematic features, namely (div, curl), (div, shear) and (curl, shear). At each pixel, we compute the orientation and magnitude of the 2-D vector corresponding to each of these pairs. The orientation is quantized into histograms and the magnitude is used for weighting, similar to SIFT. Our motivation for encoding pairs is that the joint distribution of kinematic features conveys more information than exploiting them independently.

Implementation details. The descriptor computation and parameters are similar to HOG and other popular descriptors such as MBH, HOF. We obtain 8-bin histograms for each of the three feature pairs or components of DCS. The range of possible angles is 2π for the (div,curl) pair and π for the other pairs, because the shear is always positive.

The DCS descriptor is computed for a space-time volume aligned with a trajectory, as done with the four descriptors mentioned in the previous section. In order to capture the spatio-temporal structure of kinematic features, the volume (32×32 pixels and $L = 15$ frames) is subdivided into a spatio-temporal grid of size $n_x\times n_y\times n_t$, with $n_x = n_y = 2$ and $n_t = 3$. These parameters have been fixed for the sake of consistency with the other descriptors. For each pair of kinematic features, each cell in the grid is represented by a histogram. The resulting local descriptors have a dimensionality equal to $288 = n_x \times n_y \times n_t \times 8 \times 3$. At the video level, these descriptors are encoded into a single vector representation using either BOF or the VLAD encoding scheme introduced in the next section.

6. VLAD in actions

VLAD [8] is a descriptor encoding technique that aggregates the descriptors based on a locality criterion in the feature space. To our knowledge, this technique has never been considered for action recognition. Below, we briefly introduce this approach and give the performance achieved for all the descriptors introduced along the previous sections.

VLAD in brief. Similar to BOF, VLAD relies on a codebook $C = \{c_1, c_2, ...c_k\}$ of k centroids learned by k-means. The representation is obtained by summing, for each visual word c_i, the differences $x - c_i$ of the vectors x assigned to c_i, thereby producing a vector representation of length $d\times k$,

Descriptor	Hollywood2		HMDB51	
	VLAD	BOF	VLAD	BOF
MBH	**55.1%**	54.2%	**43.3%**	39.7%
ω-MBH	**55.5%**	52.5%	**43.3%**	40.6%
ω-Trajdesc	45.5%	**51.4%**	27.8%	**32.9%**
ω-HOG	44.1%	**45.6%**	28.9%	**29.1%**
ω-HOF	**53.9%**	**53.9%**	**41.3%**	38.6%
ω-DCS	**52.5%**	50.2%	**39.1%**	35.8%
ω-DCS +ω-MBH	**56.1%**	53.1%	**45.1%**	41.2%
ω-Trajdesc + ω-HOG + ω-HOF	**59.6%**	58.5%	**47.7%**	45.6%

Table 5. Performance of VLAD with ω-Trajdesc, ω-HOG, ω-HOF descriptors and their combination.

where d is the dimension of the local descriptors. We use the codebook size, $k = 256$. Despite this large dimensionality, VLAD is efficient because it is effectively compared with a linear kernel. VLAD is post-processed using a component-wise power normalization, which dramatically improves its performance [8]. While cross validating the parameter α involved in this power normalization, we consistently observe, for all the descriptors, a value between 0.15 and 0.3. Therefore, this parameter is set to $\alpha = 0.2$ in all our experiments. For classification, we use a linear SVM and *one-against-rest* approach everywhere, unless stated otherwise.

Impact on existing descriptors. We employ VLAD because it is less sensitive to quantization parameters and appears to provide better performance with descriptors having a large dimensionality. These properties are interesting in our case, because the quantization parameters involved in the DCS and MBH descriptors have been used unchanged in Section 4 for the sake of direct comparison. They might be suboptimal when using the ω-flow instead of the optical flow on which they have initially been optimized [28].

Results for MBH and ω-MBH in Table 5 supports this argument. When using VLAD instead of BOF, the scores are stable in both the cases and there is no mixed inference as that observed in Table 3. VLAD also has significant positive influence on accuracy of ω-DCS descriptor. We also observe that ω-DCS is complementary to ω-MBH and adds to the performance. Still DCS is probably not best utilized in the current setting of parameters.

In case of ω-Trajdesc and ω-HOG, the scores are better with BOF on both the datasets. ω-HOF with VLAD improves on HMDB51, but remains equivalent for Hollywood2. Although BOF leads to better scores for the descriptors considered individually, their combination with VLAD outperforms the BOF.

7. Comparison with the state of the art

This section reports our results with all descriptors combined and compares our method with the state of the art.

Combination	Hollywood2	HMDB51
Trajectory+HOG+HOF+MBH	58.7%	48.0%
+ DCS	59.6%	49.2%
All ω-descriptors combined	**62.5%**	**52.1%**

Table 6. Combination of all five compensated descriptors using VLAD representation.

Hollywood2		HMDB51	
Ullah *et al* [26]	55.7%	Kuehne *et al* [11]	22.8%
Wang *et al* [28]	58.3%	Sadanand *et al* [21]	26.9%
*Vig *et al* [27]	60.0%	Orit *et al* [10]	29.2%
Jiang *et al* [9]	59.5%	*Jiang *et al* [9]	40.7%
Our Method	**62.5%**	Our Method	**52.1%**

Table 7. Comparison with the state of the art on Hollywood2 and HMDB51 datasets. *Vig *et al* [27] gets 61.9% by using external eye movements data. *Jiang *et al* [9] used one-vs-one multi class SVM while our and other methods use one-vs-rest SVMs. With one-against-one multi class SVM we obtain 45.1% for HMDB51.

Descriptor combination. Table 6 reports the results obtained when the descriptors are combined. Since we use VLAD, our baseline is updated that is combination of Trajectory, HOG, HOF and MBH with VLAD representation. When DCS is added to the baseline there is an improvement of 0.9% and 1.2%. With combination of all five compensated descriptors we obtain 62.5% and 52.1% on the two datasets. This is a large improvement even over the updated baseline, which shows that the proposed motion compensation and the way we exploit it are significantly important for action recognition.

The comparison with the state of the art is shown in Table 7. Our method outperforms all the previously reported results in the literature. In particular, on the HMDB51 dataset, the improvement over the best reported results to date is more than 11% in average accuracy. Jiang *el al.* [9] used a *one-against-one* multi-class SVM, which might have resulted in inferior scores. With a similar multi-class SVM approach, our method obtains 45.1%, which remains significantly better than their result. All others results were reported with *one-against-rest* approach.

On Olympic dataset we obtain mAP of **83.2%** with 'All ω-descriptors combined' and the improvement is mostly because of VLAD and ω-flow. The best reported results on this dataset are Brendel *et al* [2] (77.3%) and Jiang *et al* [9] (80.6%), which we exceed convincingly.

8. Conclusions

This paper first demonstrates the interest of canceling the dominant motion (predominantly camera motion) to make the visual motion truly related to actions, for both the trajectory extraction and descriptor computation stages. It produces significantly better versions (called compensated de-

scriptors) of several state-of-the-art local descriptors for action recognition. The simplicity, efficiency and effectiveness of this motion compensation approach make it applicable to any action recognition framework based on motion descriptors and trajectories. The second contribution is the new DCS descriptor derived from the first-order scalar motion quantities specifying the local motion patterns. It captures additional information which is proved complementary to the other descriptors. Finally, we show that VLAD encoding technique instead of bag-of-words boosts several action descriptors, and overall exhibits a significantly better performance when combining different types of descriptors. Our contributions are all complementary and significantly outperform the state of the art when combined, as demonstrated by our extensive experiments on the Hollywood 2 (the gain is +2.5% of mAP), HMDB51 (+11.4% of accuracy) and Olympic sports (+2.6% of mAP) datasets.

Acknowledgments

This work was supported by the Quaero project, funded by Oseo, French agency for innovation. We acknowledge Heng Wang's help for reproducing some of their results.

References

[1] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. IEEE *T-PAMI*, 32(2):288–303, Feb. 2010.

[2] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, Nov. 2011.

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, Sep. 2010.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, Jun. 2005.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, May 2006.

[6] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, Oct. 2005.

[7] A. Hervieu, P. Bouthemy, and L. Cadre. A statistical video content recognition method using invariant features on object trajectories. IEEE *T-CSVT*, 18(11):1533–1543, 2008.

[8] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local descriptors into compact codes. IEEE *T-PAMI*, 34(9):1704–1716, 2012.

[9] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *ECCV*, Oct. 2012.

[10] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, Oct. 2012.

[11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, Nov. 2011.

[12] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, Oct. 2003.

[13] I. Laptev, M. Marzalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, Jun. 2008.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov. 2004.

[15] M. Marzalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, Jun. 2009.

[16] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Workshop on Video-Oriented Object and Event Classification, ICCV*, Sep. 2009.

[17] R. Messing, C. J. Pal, and H. A. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, Sep. 2009.

[18] J. C. Niebles, C.-W. Chen, and F.-F. Li. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, Sep. 2010.

[19] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Jal of Vis. Comm. and Image Representation*, 6(4):348–365, Dec. 1995.

[20] G. Piriou, P. Bouthemy, and J.-F. Yao. Recognition of dynamic video contents with global probabilistic models of visual motion. IEEE *T-IP*, 15(11):3417–3430, 2006.

[21] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, Jun. 2012.

[22] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. IEEE *T-PAMI*, 19(5):530–534, May 1997.

[23] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, Oct. 2003.

[24] J. Sun, X. Wu, S. Yan, L. F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, Jun. 2009.

[25] H. Uemura, S. Ishikawa, and K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *BMVC*, Sep. 2008.

[26] M. M. Ullah, S. N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, Sep. 2010.

[27] E. Vig, M. Dorr, and D. Cox. Saliency-based space-variant descriptor sampling for action recognition. In *ECCV*, Oct. 2012.

[28] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, Jun. 2011.

[29] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, Sep. 2009.

[30] G. Willems, T. Tuytelaars, and L. J. V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, Oct. 2008.

[31] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *ICCV*, Nov. 2011.

[32] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, Jun. 2009.